**Why Sphinx is Often the Best Choice:**

1. **Highly Customizable and Versatile:**
   * Sphinx provides a wide range of extensions (like autodoc, napoleon, viewcode) that make it adaptable to various documentation needs, whether you're documenting a small script or a large-scale project.
   * You can create detailed, structured documents with sections, subsections, and cross-references, which is especially useful for complex projects.
2. **Automatic API Documentation:**
   * With the autodoc extension, Sphinx can automatically generate documentation from your Python docstrings, keeping the code and documentation in sync.
   * This reduces the manual workload and ensures that your documentation is always up-to-date with your codebase.
3. **Support for Multiple Output Formats:**
   * Sphinx can generate documentation in various formats, including HTML, PDF, ePub, and even LaTeX. This versatility makes it suitable for different distribution channels and user preferences.
4. **Integration with Read the Docs:**
   * Sphinx integrates seamlessly with Read the Docs, a popular platform for hosting open-source documentation. This allows for continuous deployment of your docs, making updates available immediately after a commit.
5. **Rich Theming and Styling Options:**
   * Sphinx offers numerous themes, such as the popular Read the Docs theme, which provides a modern, responsive design. It also allows for custom styling to match your project branding.
6. **Strong Community and Ecosystem:**
   * Sphinx has a large user community and extensive documentation itself, which means finding help, extensions, and themes is easier. The ecosystem around Sphinx is mature, making it a reliable choice.
7. **Support for reStructuredText (reST) and Markdown:**
   * While primarily using reStructuredText, Sphinx also supports Markdown with the right extensions, allowing you to choose the format you're most comfortable with.
8. **Extensive Cross-Referencing and Linking Capabilities:**
   * Sphinx excels in cross-referencing, which is essential for creating detailed and interconnected documentation, especially for APIs and libraries.

**When Sphinx Might Not Be the Best Fit:**

* **Learning Curve**: Sphinx can have a steeper learning curve compared to simpler tools like MkDocs.
* **Simple Projects**: For very small or markdown-only projects, MkDocs might be quicker and easier to set up.